International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 7 Issue 3, May-June 2023 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Lane Detection and Object Detection

Harshada Annasaheb Nikam, Pranav Ashokrao Survase, Ketan Patole, Mansi Sanjay Gore

Department of Information Technology (IT) Engineering, Siddhant College of Engineering, Pune, Maharashtra, India

ABSTRACT

Every person in this world is concerned about being safe. Increasing safety and reducing road accidents, thereby saving lives are one of great interest in the context of Advanced Driver Assistance Systems. Among the complex and challenging tasks of future road vehicles is road lane detection or road boundaries detection. In driving assistance systems, obstacle detection especially for moving object detection is a key component of collision avoidance and also detect signs boards. The most frequently used principal approach to detect road boundaries and lanes using vision system on the vehicle. In proposed system, we can implement of lane Object detection and also detect Signs board with the help of AI and Deep learning Technique. Input as Dataset to trained using CNN algorithm and creat beast model. Using these model to classify or detect output.

How to cite this paper: Harshada Annasaheb Nikam | Pranav Ashokrao Survase | Ketan Patole | Mansi Sanjay Gore "Lane Detection and Object Detection" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-3, June 2023, pp.365-381, URL: www.ijtsrd.com/papers/ijtsrd57393.pdf

IJTSRD57393

1. INTRODUCTION

Everyone on the planet is concerned about their safety. People who travel from one location to another hope to arrive safely. Without any unexpected events, such as traffic accidents, which may occur while travelling. Using improved driving assistances, we can avoid road accidents. In most populated countries, vehicle collisions are still the major cause of accidental death and injury.

People are investing a lot of money on advancements in driving tactics that ensure safety in order to boost safety and reduce road accidents. Technology encourages people to think more about how to improve safety and save lives. Road lanes detection or boundaries detection (white and black lines on roads) and Obstacles detection (cars, pedestrians, trees, etc) are complex and challenging tasks for future road vehicles, especially for moving object detection and detecting road signs like (Speed, Right, Left signs) is a key component of collision avoidance in driving assistance systems.

The AI technology uses a camera through capture the front view, then utilizes a CNN algorithms to detect lanes, objects, and road signs. To detect the lanes, objects, and road signs, a diverse methodology is applied. Vehicles equipped with sophisticated systems such as road lane detection and obstacle detection are safer, which is critical in reducing the number of people killed or injured in car accidents. The use of a vehicle's visual system is one of the most common methods of detection.

1.1. MOTIVATION

Safety-critical computer vision-based on-road systems demand excellent real-time performance, and these systems must work consistently in a variety of road, weather, and traffic circumstances. The availability of a large variety of advanced technologies in software component integration provides a number of unique issues in the design and developments of computer vision-based improved driver assistance systems for intelligent vehicles .A wide selection of open source libraries, such as OpenCV and Python, provide adequate motivation to design, implement, and test road lane detection Object detection, as well as sign board detection methods.

1.2. PROBLEM DEFINITION

Lane object Detection Systems which work with the intention to recognize the lane borders and object, Sign Board on road and further prompts the driver if he switches and moves to erroneous lane markings. To detect object on road (like cars, peoples, animals) with the help of computer vision technology.

2. LITERATURE SURVEY

2.1. STUDY OF RESEARCH PAPER

Hui Lv, Chao Liu, Xiaowei Zhao, Chunyan Xu, Zhen Cui, Jian Yang [1]., we proposed by a present a deep neural network based real-time integrated framework to detect objects, lane markings, and drivable space using a monocular camera for advanced driver assistance systems. The object detection framework detects and tracks objects on the road such as cars, trucks, pedestrians, bicycles, motorcycles, and traffic signs. The lane detection framework identifies the different lane markings on the road and also distinguishes between the ego lane and adjacent lane boundaries.

Mehdi FENICHE, Tomader MAZRI [2] are ex planing that Nowadays, Lane detection and tracking modules are considered as central requirements in every Intelligent Transportation System (ITS) development. The extracted lane information could be used in several smart applications for lane keeping systems, lane departure warning and avoiding collisions with other vehicles. In this proposed work, we are presenting, reviewing and comparing the different vision based algorithms used for detecting road lanes in autonomous vehicles.

Jung Uk Kim and Yong Man Ro [3] author proposed a system, Object detection became one of the major fields in computer vision. In object detection, object classification and object localization tasks are conducted. Previous deep learning based object detection networks perform with feature maps generated by completely shared networks. However, object classification focuses on the most discriminative object part of the feature map. Whereas, object localization requires a feature map that is focused on the entire area of the object.

Ziqiang Sun [4], are evaluating Lane detection is a key reference for traffic safety .In the intelligent transportation system, lane line is the most important traffic sign in road traffic, which can restrain and guarantee the running of vehicles, so as to maintain the safety highly secured [1-2]. Machine vision is considered to be effective and simple during the process of lane detection. Lane line detection and identification has become a basic and necessary functional module in the field of vehicle safety and intelligent vehicle navigation, which can not only reduce the occurrence of traffic accidents, but also provide help for in depth research on intelligent traffic. The lane detection technology based on machine vision also has the advantages of low cost and strong versatility and has been widely used.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1. PROJECT SCOPE

lane and object is a part of a road that is designed for a single line of vehicles and is used to identify an object and then road signs. It is used to guide and control drivers, as well as to reduce traffic congestion. As the incidence of traffic accidents rises, so does concern about the nature of the accidents. Human mistake is the most common cause. As a result, lane and object detection systems, as well as sign board detection systems, are being developed as a means of supporting the driver

3.2. ASSUMPTIONS AND DEPENDENCIES

Assumptions:

We have used Python Technique. Input as lane Object and road sign images dataset. Dependencies:

We have used python libraries like Tensorflow, keras, opencv and Tkinter. Output to detect Lane Object and also Detect road sign.

3.3. FUNCTIONAL REQUIREMENTS

Functional Specification:

- The application is user friendly.
- It provides an easy interface to user.
- The accessibility or response time of the application should be fast.

Dependency and Constraints:

- End User application will be developed in Windows OS.
- All scripts shall be written in Python.
- Application design pattern shall be Singleton.

3.4. NON FUNCTIONAL REQUIREMENT

3.4.1. Performance Requirements

The performance of the functions and every module must be well. The overall performance of the software will enable the users to work recently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast Safety Requirement

The application is designed in modules where errors can be detected and xedeasily.

This makes it easier to install and update new functionality if required.

3.4.2. Safety Requirement

The application is designed in modules where errors can be detected and fixed easily.

This makes it easier to install and update new functionality if required.

3.4.3. Software Quality Attributes

Our software has many quality attribute that are given below:-

Adaptability: This software is adaptable by all users.

Availability: This software is freely available to all users. The availability of the software is easy for everyone.

Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

Reliability: The performance of the software is better which will increase the reliability of the Software.

User Friendliness: Since, the software is a GUI application; the output generated is much user friendly in its behavior.

Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.

Security: Users are authenticated using many security phases so reliable security is provided.

Testability: The software will be tested considering all the aspects.

3.5. SYSTEM REQUIREMENT

3.5.1. Hardware Interfaces:

- Hardware : intel core
- Speed : 2.80 GHz
- RAM : 8GB
- HardDisk : 500 GB
- Key Board: Standard Windows Keyboard

3.5.2. Software Interfaces

- Operating System: Windows 10(64 Bit)
- IDE: Spyder
- Programming Language : python version 3.7,3.8
- Libraries: Tensor Flow, OpenCv, Keras,Numpy

3.6. SYSTEM IMPLEMENTATION PLAN

The System Implementation plan table, shows the overall schedule of tasks compilation and time duration required for each task.

Sr. No.	Name/Title	Start Date	End Date
1	Preliminary Survey		
2	Introduction and Problem Statement		
3	Literature Survey		
4	Project Statement		
5	Software Requirement And Specification		
6	System Design		
7	Partial Report Submission		
8	Architecture Design		
9	Implementation		
10	Deployement		
11	Testing		
12	Paper Publish		
13	Report Submission		

4. SYSTEM DESIGN

4.1. SYSTEM ARCHITECTURE

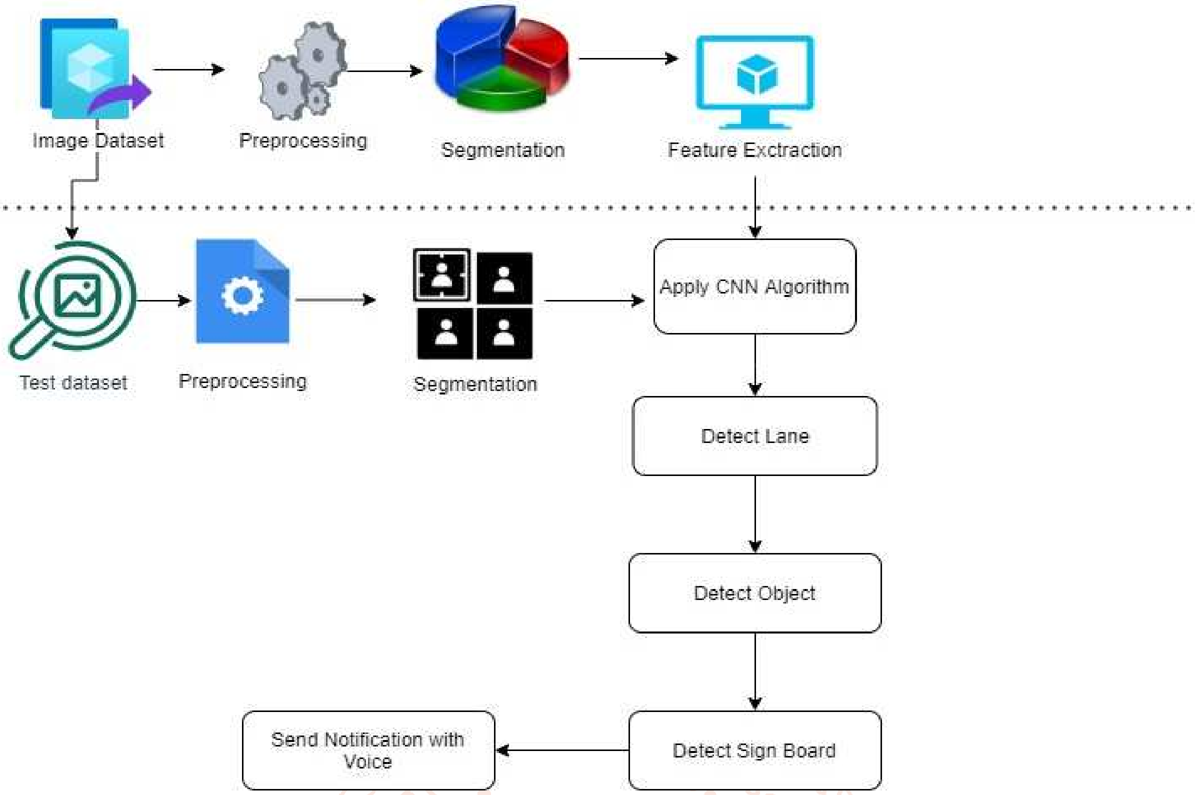


Figure 4.1: system Architecture

4.1.1. Data Flow Diagram

In Data Flow Diagram, we Show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system, In DFD1 we show actual input and actual output of system input of our system is text or image and output is rumor detected

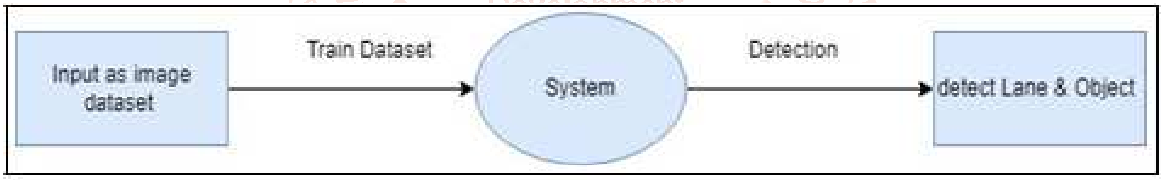


Figure 4.2: Data Flow diagram

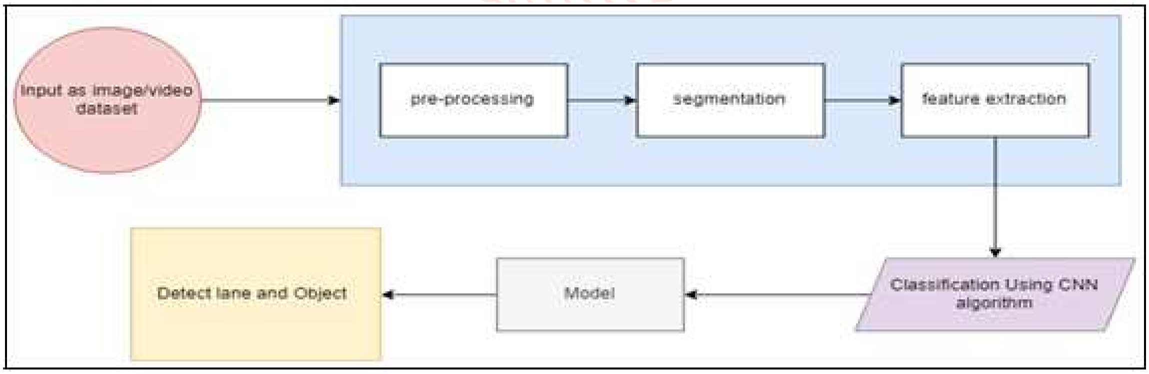


Figure 4.3: Data Flow diagram

4.2. UML DIAGRAMS

Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent, although optimally it should be used in process that is use case driven, architecture-centric, iterative, and incremental. The Number of UML Diagram is available.

Activity Diagram.
Sequence Diagram.
Use case Diagram.

Activity Diagram

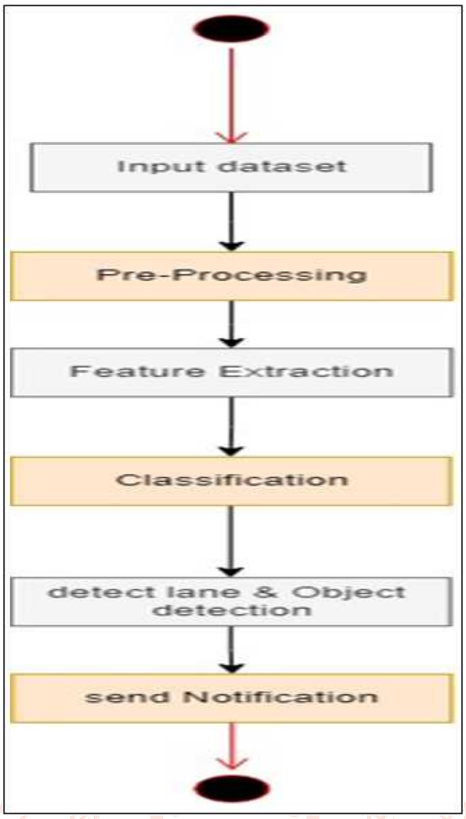


Figure 4.4: Activity Diagram

Sequence Diagram

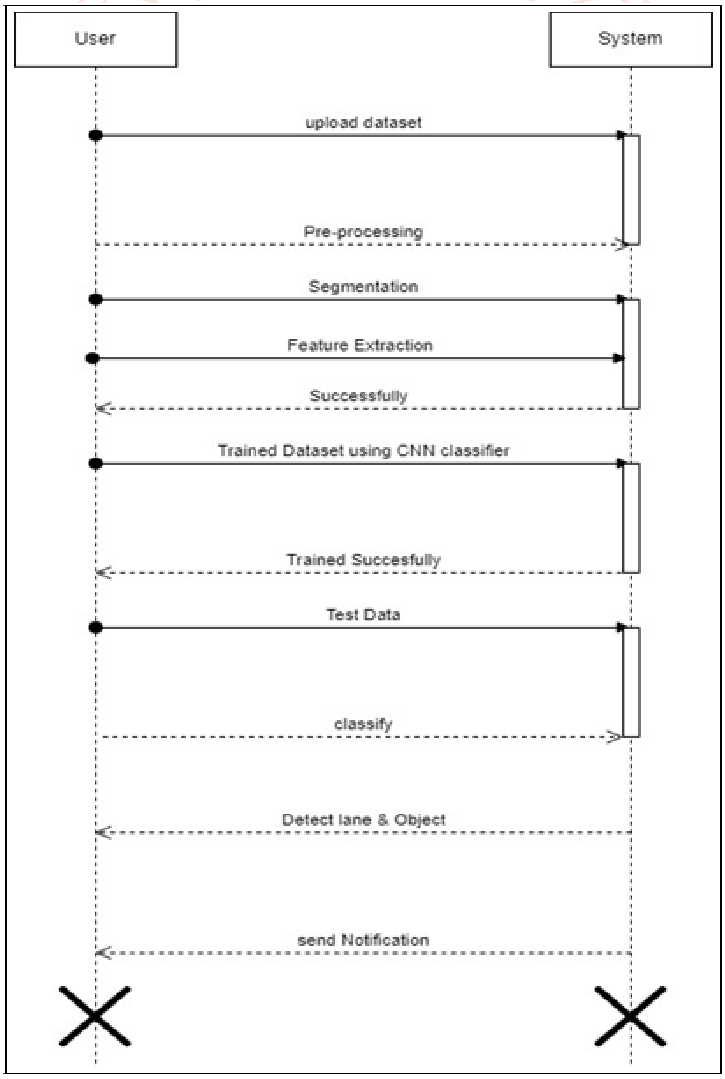


Figure 4.5: Sequence Diagram

Use case Diagram

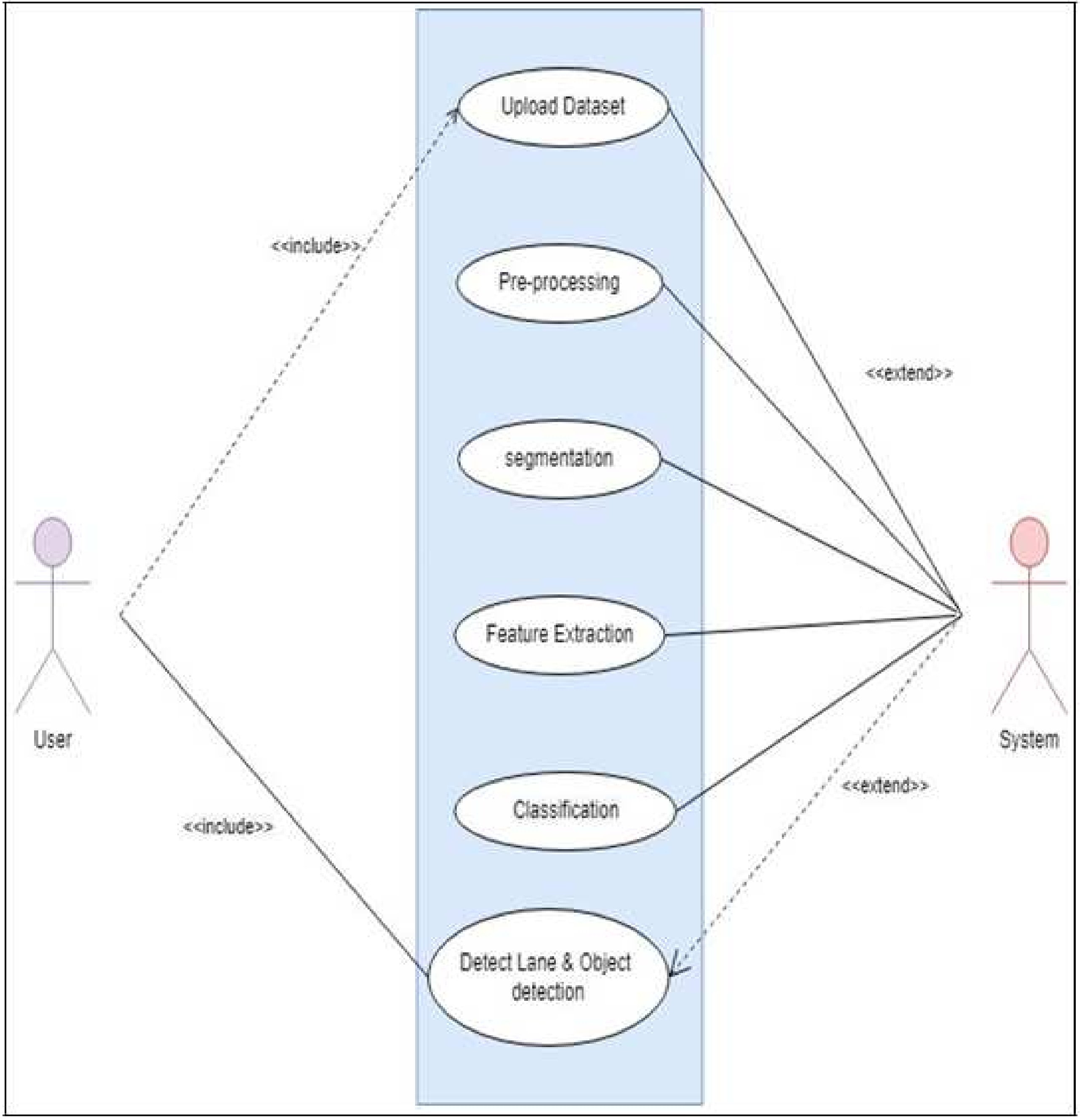


Figure 4.6: Use case Diagram

4.2.1. Analysis Models: Waterfall Model

SDLC Models stands for Software Development Life Cycle Models. In this article, we explore the most widely used SDLC methodologies such as Agile. Each software development life cycle model starts with the analysis, in which the Also, here are defined the technologies used in the project, team load. One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. There is no one single SDLC model. They are divided into main groups, each with its features and weaknesses.

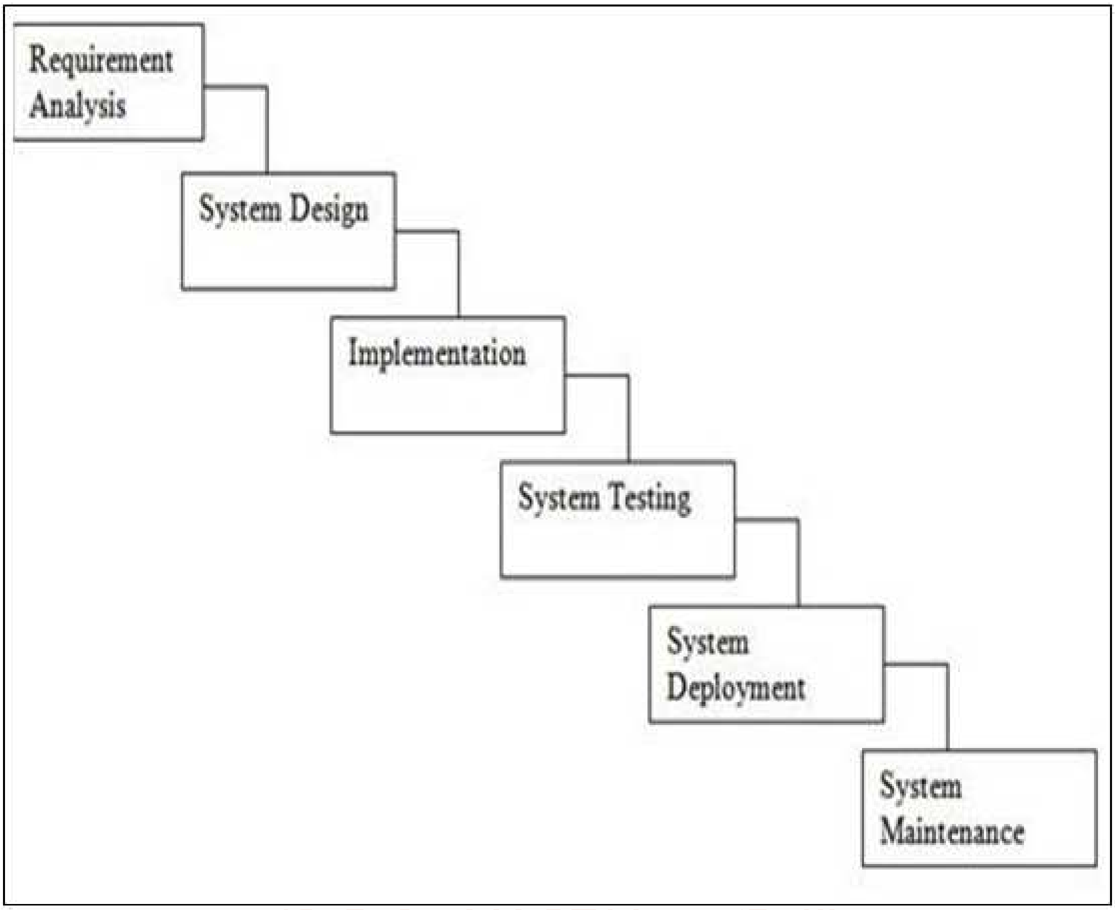


Figure 4.7: Waterfall Model

1. Requirement Analysis - Requirement Analysis is the most important and necessary stage in SDLC. The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry. Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage. Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

2. System Design - The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

3. Implementation - In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

4. Testing - After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage. During this stage, unit testing, integration testing, system testing, acceptance testing are done.

5. Deployment - Once the software is certified, and no bugs or errors are stated, then it is deployed. Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment. After the software is deployed, then its maintenance begins.

6. Maintenance - Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time. This procedure where the care is taken for the developed product is known as maintenance.

5. Project Plan

5.1. PROJECT ESTIMATES

5.1.1. Reconciled Estimates

Cost Estimate The project cost can be found using any one of the model. COCOMO 1 Model COCOMO-2 Model Model -1: The basic COCOMO model computes software development efforts as a function of program size expressed in estimated lines of code. Model-2: The intermediate COCOMO model computes software development efforts as a function of program size and a set of cost drivers that include subjective assessment of the product, hardware, personnel, project attributes Model-3: The advanced COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost drivers impact on each step of the software engineering process. Following is the basic COCOMO -2 model.

1. Software Project A(b) B(b) C(b) D(b)
2. Organic 2.4 1.05 2.5 0.38
3. Semi-detached 3.0 1.22 2.5 0.35
4. Embedded 3.6 1.20 2.5 0.32 25

The basic COCOMO -2 model equations take form
E=A(b)KLOCB(b)

D=C(b)ED(b)

Where E is the effort applied in person months. D is development time in chronological month. KLOC is estimated number of delivered lines of code for the project. This project can be classified as Semidetached software project. The rough estimate of number of lines of this project is 9.072k.Applying the above formula

E=3.0*(9.072)1.22

5.1.2. Project Resources

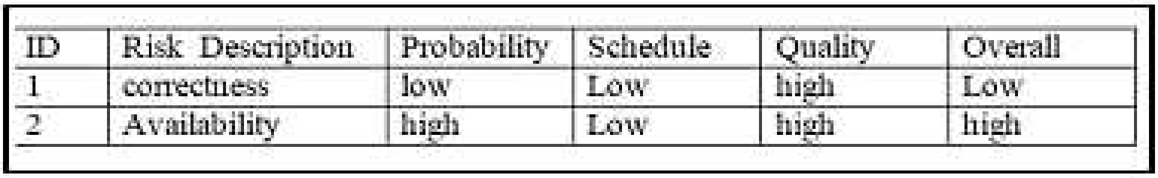
Well configured Laptop, Spyder software, 2 GHZ CPU speed, 2 GB RAM, Internet connection.

5.2. RISK MANAGEMENT

5.2.1. Risk Identification

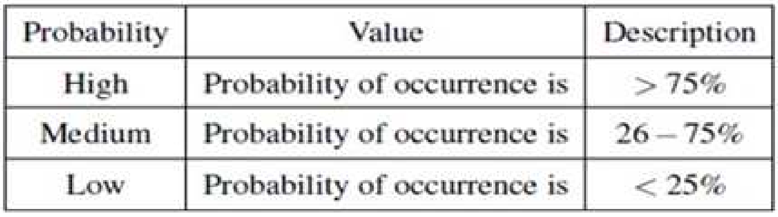
1. Have top software and customer managers formally committed to support the project?
   Ans-Not applicable.
2. Are end-users enthusiastically committed to the project and the system/product tobe built?
   Ans-Not known at this time.
3. Are requirements fully understood by the software engineering team and its customers?
   Ans-Yes
4. Have customers been involved fully in the definition of requirements?
   Ans-Not applicable
5. Do end-users have realistic expectations?
   Ans-Not applicable
6. Does the software engineering team have the right mix of skills?
   Ans-yes
7. Are project requirements stable?
   Ans-Not applicable

5.2.2. Risk Analysis



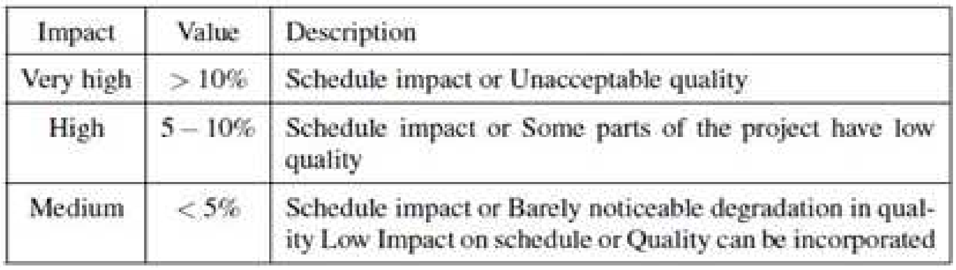
ID	Risk Description	Probability	Schedule	Quality	Overall
1	correctness	low	Low	high	Low
2	Availability	high	Low	high	high

Figure 5.1: Risk Table



Probability	Value	Description
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 – 75%
Low	Probability of occurrence is	< 25%

Figure 5.2: Risk Probability definitions?



Impact	Value	Description
Very high	> 10%	Schedule impact or Unacceptable quality
High	5 – 10%	Schedule impact or Some parts of the project have low quality
Medium	< 5%	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Figure 5.3: Risk Impact definitions?

5.2.3. Risk Mitigation Risk Monitoring and Risk Management

1. Risk Mitigation:

If a software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation. To mitigate this risk, you would develop a strategy for reducing turnover. Among the possible steps to be taken are:

- Meet with current staff to determine causes for turnover (e.g., poor working condi-tions, low pay, competitive job market).
- Mitigate those causes that are under your control before the project starts.
- Once the project commences, assume turnover will occur and develop techniquesto ensure continuity when people leave.
- Organize project teams so that information about each development activity iswidely dispersed.
- Define work product standards and establish mechanisms to be sure that all modulesand documents are developed in a timely manner.
- Conduct peer reviews of all work.
- Assign a backup staff member for every critical technologist.

2. Risk Monitoring

As the project proceeds, risk-monitoring activities commence. The project manager monitors factors that may provide an indication of whether the risk is becoming more or less likely. In the case of high staff turnover, the general attitude of team members based on project pressures, the degree to which the team has jelled, interpersonal relationships among team members, potential problems with compensation and benefits, and the availability of jobs within the company and outside it are all monitored.

3. Risk Management

Risk management and contingency planning assumes that mitigation efforts have failed and that the risk has become a reality. Continuing the example, the project is well under way and a number of people announce that they will be leaving. If the mitigation strategy has been followed, backup is available, information is documented, and knowledge has been dispersed across the team. In addition, you can temporarily refocus resources (and readjust the project schedule) to those functions that are fully staffed, enabling newcomers who must be added to the team to "get up to speed." Those individuals who are leaving are asked to stop all work and spend their last weeks in "knowledge transfer mode." This might include video based knowledge capture, the development of "commentary documents or Wikis," and/or meeting with other team members who will remain on the project.

5.3. PROJECT SCHEDULE

5.3.1. Project Task Set

Major Tasks in the Project stages are:

- Task 1: correctness
- Task 2: availability
- Task 3: integrity

5.3.2. Task Network

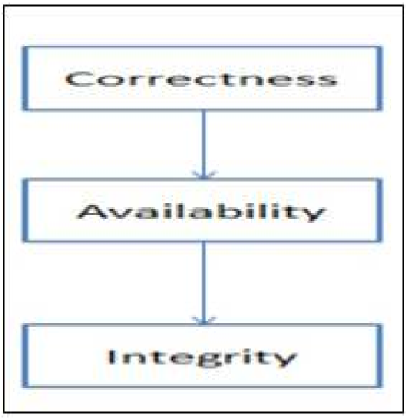


Figure 5.4: Task Network

5.3.3. Timeline Chart

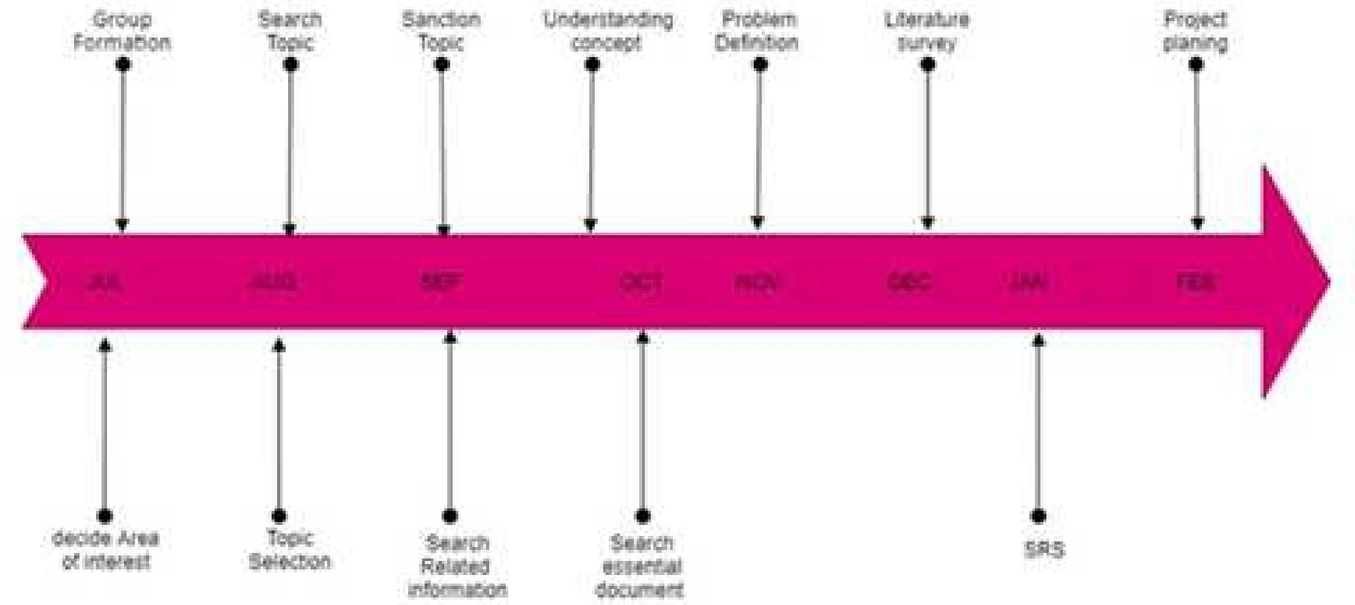


Figure 5.5: Timeline Chart

5.4. TEAM ORGANIZATION

5.4.1. Team Structure

The team structure for the project is identified. There are total 2 members in our team and roles are defined. All members are contributing in all the phases of project.

Sr. No.	Name/Title	Start Date	End Date
1	Preliminary Survey		
2	Introduction and Problem Statement		
3	Literature Survey		
4	Project Statement		
5	Software Requirement And Specification		
6	System Design		
7	Partial Report Submission		
8	Architecture Design		
9	Implementation		
10	Deployement		
11	Testing		
12	Paper Publish		
13	Report Submission		

5.4.2. Management reporting and communication:

Well planning mechanisms are used for progress reporting and inter/intra team communication are identified as per requirements of the project.

6. Project Implementation

6.1. OVERVIEW OF PROJECT MODULES

In this chapter we are going to have an overview about how much time does it took to complete each task like- Preliminary Survey Introduction and Problem Statement, Literature Survey, Project Statement, Software Requirement and Specification, System Design, Partial Report Submission, Architecture Design, Implementation, Deployment, Testing, Paper Publish, Report Submission. This chapter also gives focus on stakeholder list which gives information about project type, customer of the proposed system, user and project member who developed the system.

6.2. TOOLS AND TECHNOLOGIES USED

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a ”batteries included” language due to its comprehensive standard library.

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and ”Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release.”[30] No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his ”permanent vacation” from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member ”Steering Council” to lead the project.

Anaconda: Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages[citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation[citation needed]. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

Spyder

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your own software.

Features

➢ **Editor**

Work efficiently in a multi-language editor with a function/class browser, realtime code analysis tools (pyflakes, pylint, and pycodestyle), automatic code completion (jedi and rope), horizontal/vertical splitting, and go-to-definition.

➢ **Interactive console**

Harness the power of as many I Python consoles as you like with full workspace and debugging support, all within the flexibility of a full GUI interface. Instantly run your code by line, cell, or file, and render plots right in line with the output or in interactive windows.

➢ **Documentation viewer**

Render documentation in real-time with Sphinx for any class or function, whether external or user-created, from either the Editor or a Console.

6.3. ALGORITHM

Convolutional Neural Network:

Convolutional neural network is one of the main categories to do image recognition, image classification, object detection widely used.

CNN image classification takes the input image, process it and classify it. Computer sees an input image as array of pixels depends on the image resolution.(h*w*h)

CNN is a widely-used image recognition model that has been shown to attain greater than 78.1 percent accuracy on the Image Net dataset. The model is the conclusion of many ideas developed by many researchers over the years. An 256x256x3 input representing a visual field of 256 pixels and 3 color (RGB) channels. Five convolution layers, with a few interspersed max pooling operations. Successive stacks of “CNN Models”. A Soft Max output layer at the end at an intermediate output layer just after the mixed layer. Steps involved in CNN are Convolution layer in CNN is performed on an input image using a filter. Relu (Rectified Linear Unit) which simply converts all of the negative values to 0 and keeps the positive values the same. Pooling layer is used to reduce the spatial size of the Convolved Feature. They are of two types such as Max Pooling and Average Pooling. Fully Connected layers in a neural networks is a layer where all the inputs from one layer are connected to every activation unit of the next layer. These networks are commonly trained under a log loss (or cross-entropy) system, giving a nonlinear variant of multinomial logistic regression.

2 CNN LAYERS:

Conv2D: It is the layer to convolve the image into multiple images activation is the activation function.

MaxPooling2D: It is used to max pool the value from the given size matrix and same is used for the next 2 layers.

Dropout: It is used to avoid over fitting on the dataset and dense is the output layer contains only one neuron which decide to which category image belongs.

Fully Connected: It has neurons that are fully connected to the neurons in previous layer. This FC layer is often kept as the final layer of a CNN with "SOFTMAX" as its activation function for multi-class classification problems. The FC layer is responsible to predict the final class or label of the input image. Thus, it has an output dimension of [1x1xN] where N denotes the number of classes or labels considered for classification.

Epochs: It tells us the number of times model will be trained in forward and backward pass.

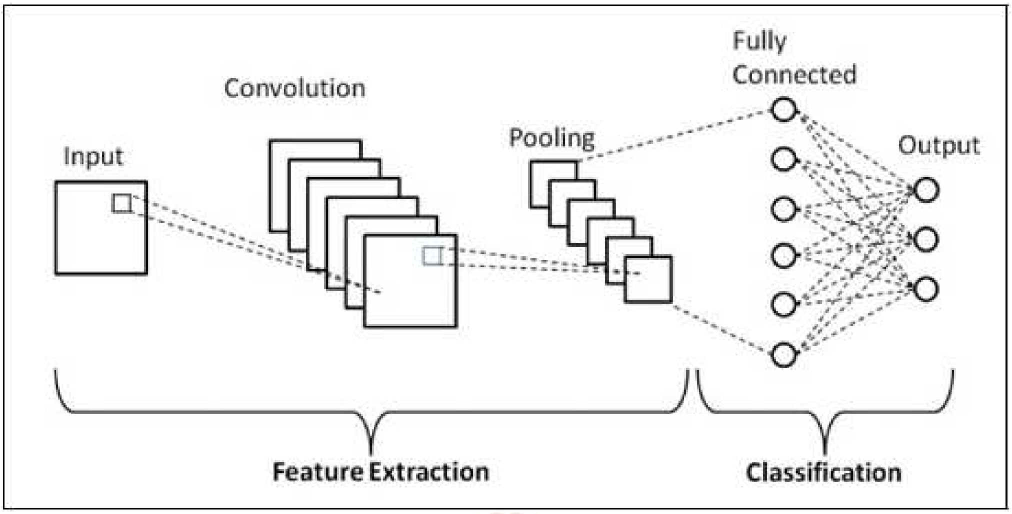


Figure 6.1: CNN Algorithm

CNN's were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

Convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

Hidden Layer: The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

7. Software Testing

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software testing can also be stated as the process of validating and verifying that a software program or application or product:

1. Meets the business and technical requirements that guided
2. Works as expected;
3. Can be implemented with the same characteristics

7.1. TYPES OF TESTING

7.1.1. Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units.

7.1.2. Regression Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output.

7.1.3. Smoke Testing

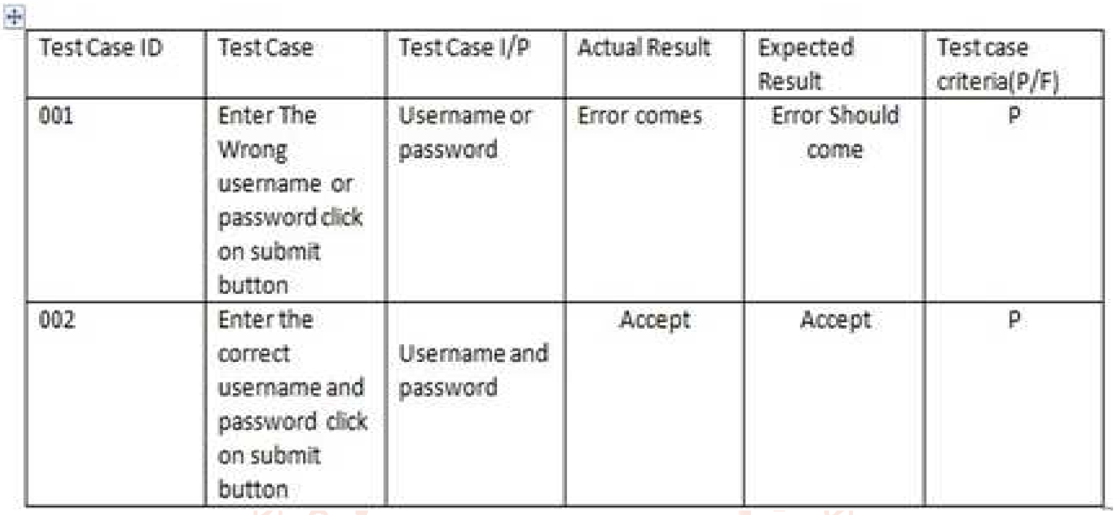
Very time new module is added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program.

7.1.4. System Testing

In this software is tested such that it works fine for different operating system.It is covered under the black box testing technique.

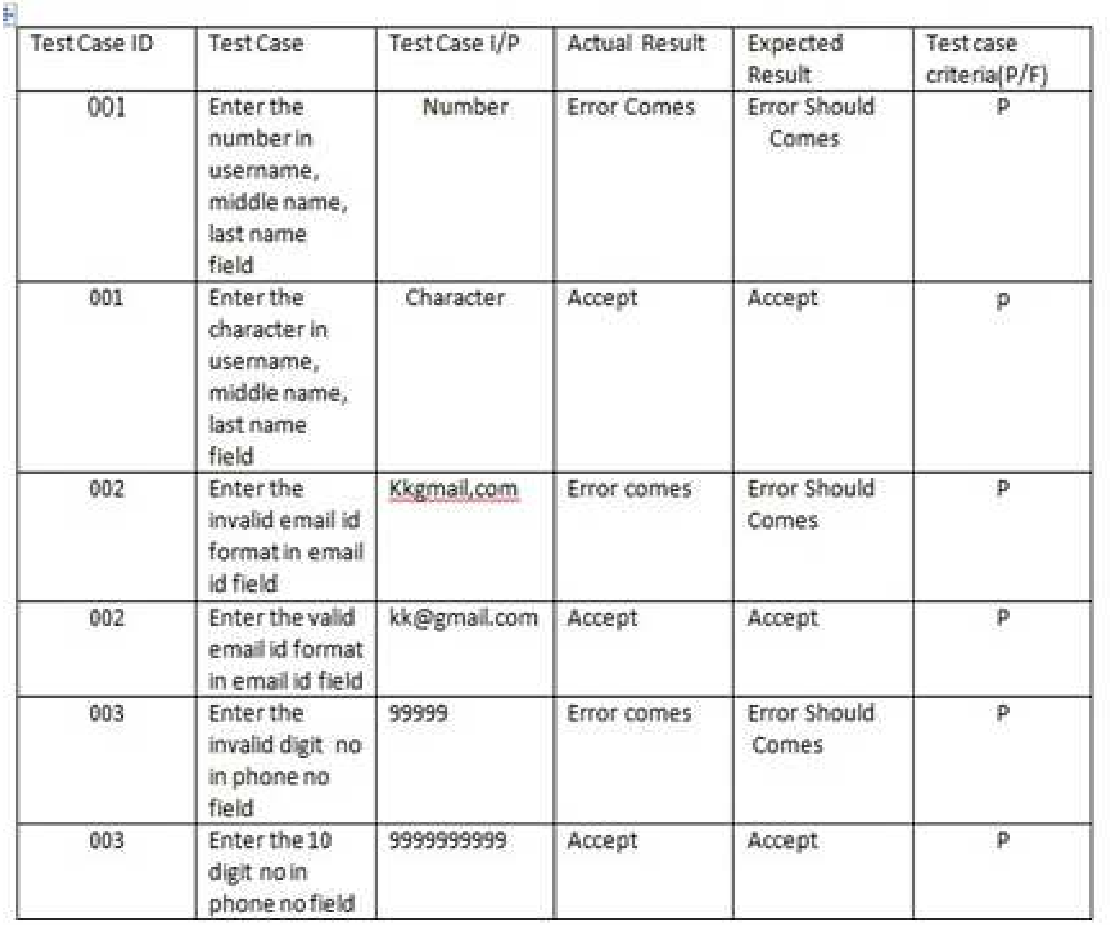
7.1.5. Test Cases

Login test case

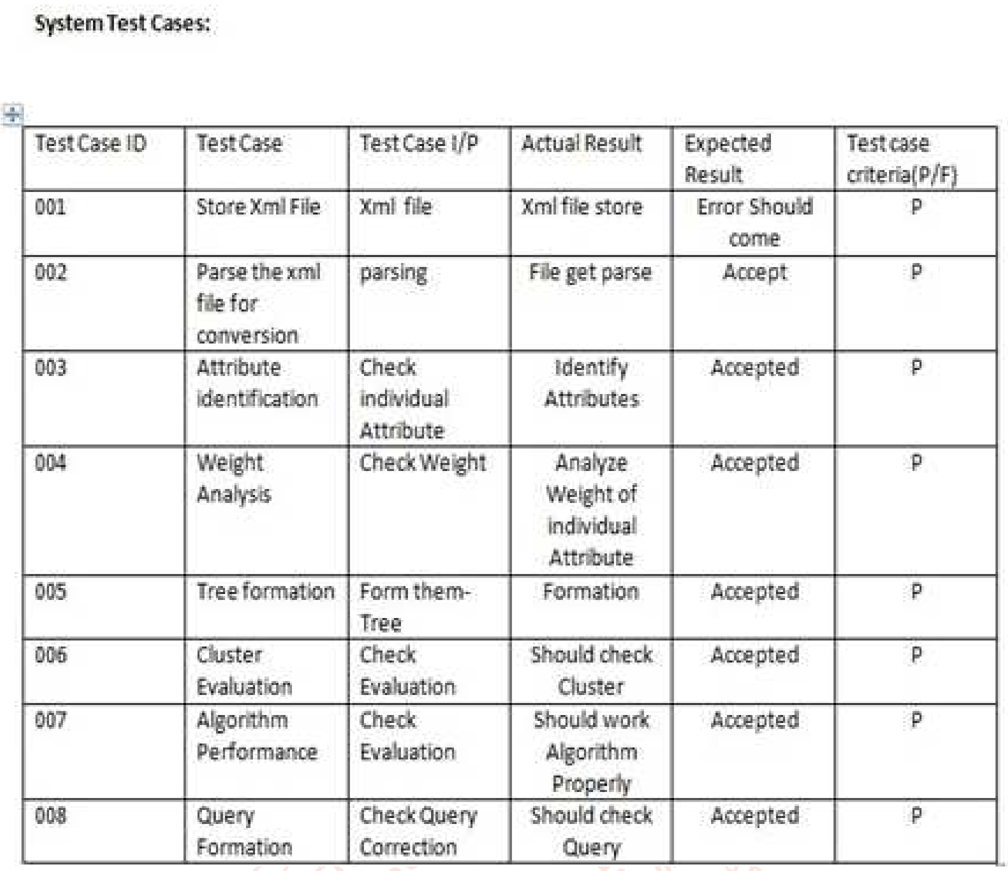


Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Enter The Wrong username or password click on submit button	Username or password	Error comes	Error Should come	P
002	Enter the correct username and password click on submit button	Username and password	Accept	Accept	P

Registration test case



Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Enter the number in username, middle name, last name field	Number	Error Comes	Error Should Comes	P
001	Enter the character in username, middle name, last name field	Character	Accept	Accept	p
002	Enter the invalid email id format in email id field	Kkgmail.com	Error comes	Error Should Comes	P
002	Enter the valid email id format in email id field	kk@gmail.com	Accept	Accept	P
003	Enter the invalid digit no in phone no field	99999	Error comes	Error Should Comes	P
003	Enter the 10 digit no in phone no field	9999999999	Accept	Accept	P



System Test Cases:

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test case criteria(P/F)
001	Store Xml File	Xml file	Xml file store	Error Should come	P
002	Parse the xml file for conversion	parsing	File get parse	Accept	P
003	Attribute identification	Check individual Attribute	Identify Attributes	Accepted	P
004	Weight Analysis	Check Weight	Analyze Weight of individual Attribute	Accepted	P
005	Tree formation	Form them-Tree	Formation	Accepted	P
006	Cluster Evaluation	Check Evaluation	Should check Cluster	Accepted	P
007	Algorithm Performance	Check Evaluation	Should work Algorithm Properly	Accepted	P
008	Query Formation	Check Query Correction	Should check Query	Accepted	P

8. OTHER SPECIFICATION

8.1. ADVANTAGES

1. System To detect easily lane and object board.
2. Time Consuming and provide safety to driver.
3. Very help system to self-driver.

8.2. DISADVANTAGES

1. One disadvantage crash or damage camera then system is not working because system is also depend on camera.

8.3. APPLICATION

1. River Assistance.
2. Lane Departure warning.
3. Lane-keeping systems.
4. Object analyze and detect.

9. Conclusion and Future Work

9.1. CONCLUSION

We have proposed a method for detecting and tracking multiple lanes Object by effectively fusing both frame-by-frame lane detection and temporal dependencies over the sequence of frames, using a video taken by a single dash-cam. And also detect sign board using CNN algorithm.

9.2. FUTURE WORK

1. The object recognition system can be applied in the area of surveillance system,face recognition,fault detection, character recognition etc. The objective of this thesis is to develop an object recognition system to recognize the 2D and 3D objects in the image. The performance of the object recognition system depends on the features used and the classifier employed for recognition.
2. Future work will include vehicle departure warnings and pedestrian detection, orlane markings, which will effectively help drivers improve driving safety
3. Bigger dataset can be used for executing this project in a more precise way. Instead of SVM mode, we can use CNN for classification of traffic signs. There can be a better performance with high frame rate. Different approaches like YOLO or SSD can also be used. YOLO (You Only Look Once) is a CNN algorithm that uses single network for the input image, thus, creating bounding box and probability of the features across image for detection purpose.

Annexure A

NP-Hard NP-Complete:
What is P?

- P is set of all decision problems which can be solved in polynomial time by a deterministic.
- Since it can be solved in polynomial time, it can be verified in polynomial time.
- Therefore P is a subset of NP.

P: To identify road condition or road survey requires more man power, time and money. To resolve these problems we need effective system.

What is NP?
NP means we can solve it in polynomial time if we can break the normal rules of step-by-step computing.

What is NP Hard?
A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem. NP-hard therefore means ”at least as hard as any NP-problem,” although it might, in fact, be harder.

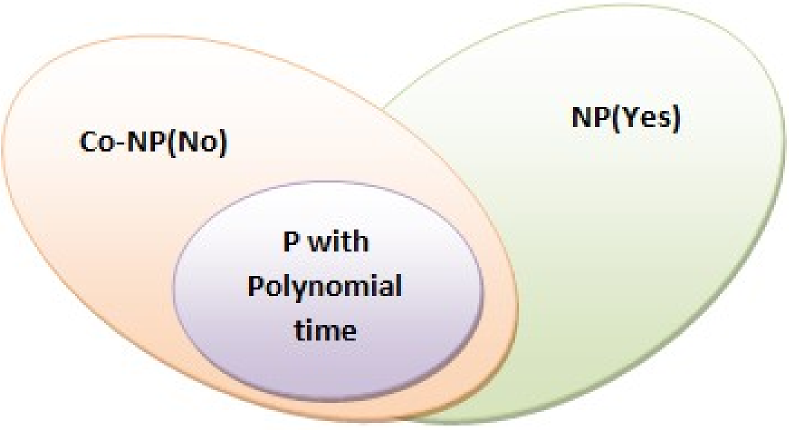


Figure A.1: P Problem

NP-Hard:
Propose system analyze the road condition and road surface. It identify bad road patches and gives notification to navigation system. For that we used inbuilt accelerometer sensor and gyroscope sensor. To improve the system result we use decision tree algorithm. Propose system has self-managing database which collect data from vehicle drivers android smart phones. This data update in real time periodically. Application utilizes this data to inform other application users about road condition.

So here in this case the ‘P’ problem is NP hard.
i.e. P=NP-Hard
What is NP-Complete?

- Since this amazing ”N” computer can also do anything a normal computer can, we know that ”P” problems are also in ”NP”.
- So, the easy problems are in ”P” (and ”NP”), but the really hard ones are *only* in ”NP”, and they are called ”NP-complete”.

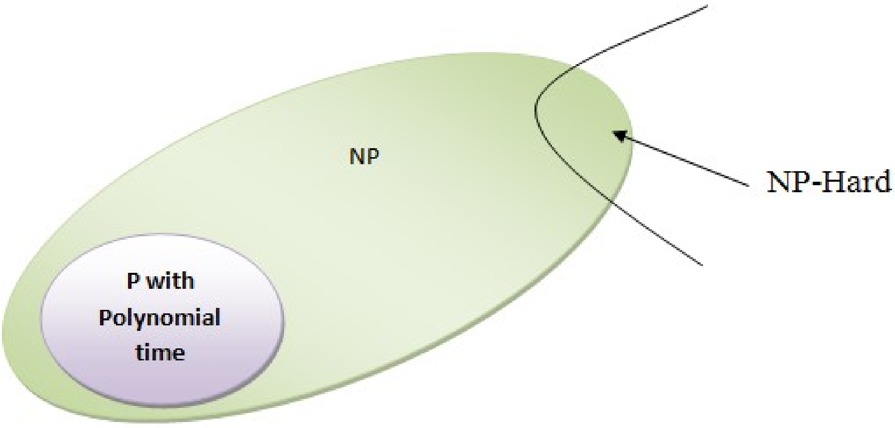


Figure A.2: NP Problem

- It is like saying there are things that People can do (”P”), there are things that Super People can do (”SP”), and there are things *only* Super People can do (”SP-complete”).

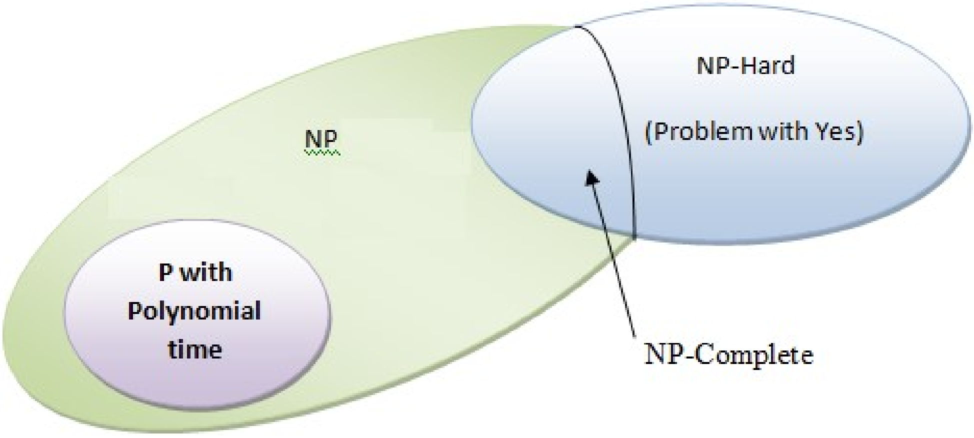


Figure A.3: NP Complete Problem

NP-Complete:
We have used inbuilt mobile sensor to identify road conditions.

Hence the ‘P’ is NP-Complete in this case.

10. References

[1] “Machine vision-based coin separator and counter” by Prashanna Rangan R, STM Journals, STM Journals, Volume 5, Issue 1, 2018

[2] “Coin counting and sorting machine” by Prof. Anupa Kavale, Prof.Shraddha Shukla, Prof. Prachi Bramhe, 2019, 9th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing [ICETET-SIP-19]

[3] “Coin detection and recognition using neural networks” by S. Mohamed mansoor roomi, R. B. Jayanthi rajee, 2015, International Conference on Circuit, Power and Computing Technologies[ICCPCT]

[4] “Study on Automatic sorting and counting machine for coin” by Liai Pan, and Qiulei Du, vols 427 - 429 (2013) pp 872-875 (2013), Trans Tech Publications, Switzerland

[5] “Automatic South African Coin Recognition through Visual Template Matching” by Tharish Sooruth, MandlenKosi V. Gwetu, 2018, IEEE

[6] ”Indian Coin Recognition System of Image Segmentation by Heuristic Approach and Houch Transform (HT)” by C. M. Velu and P. Vivekanandhan, Int. J. Open Problems Compt.Math., Vol. 2, No.2, June 2009.

[7] ”Indian Coin Detection and Sorting using SIFT Algorithm” by Rohan.S.Prabhu, Sahil.G. Khorjuvekar, Akshat.G.Poi, Vaibhav Naik, IJSTE-International Journal of Science Technology Engineering, Volume 2, Issue 10, April 2016.

[8] Harveen Kaur and Neetu Sharma, “Modified coin identification using Neural Network”, IJETAE 2014

[9] Saranya das. Y. M and R. Pugazhenthi, “HarrisHessian Algorithm for coin Apprehension”, IJARCET 2013

[10] Tushar N. Nimbhorkar and M. M. Bartere, “Coin Recognition by using Artificial Neural Network”, IJCITB 2013